



Diagnosis for Control and Decision Support for Autonomous Vehicles

Blanke, Mogens; Hansen, Søren; Rufus Blas, Morten

Published in:
Complex Systems

Link to article, DOI:
[10.1007/978-3-319-28860-4_1](https://doi.org/10.1007/978-3-319-28860-4_1)

Publication date:
2016

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Blanke, M., Hansen, S., & Rufus Blas, M. (2016). Diagnosis for Control and Decision Support for Autonomous Vehicles. In G. M. Dimirovski (Ed.), *Complex Systems: Relationships between Control, Communications and Computing* (pp. 3-37). Springer. Studies in Systems, Decision and Control Vol. 55 https://doi.org/10.1007/978-3-319-28860-4_1

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Diagnosis for Control and Decision Support for Autonomous Vehicles

Mogens Blanke^{1,2}, Søren Hansen¹, and Morten Rufus Blas³

¹Automation & Control Group, Department of Electrical Engineering,
Technical University of Denmark, Kgs. Lyngby, Denmark

²AMOS CoE, Institute for Technical Cybernetics,
Norwegian Univ. of Science and Technology, Trondheim, Norway

³CLAAS Agrosystems, Møllevej 11, Nivå, Denmark
{mb,sh}@elektro.dtu.dk; {rufus.blas}@claas.com

Abstract. Diagnosis and, when possible, prognosis of faults are essential for safe and reliable operation. The area of fault diagnosis has emerged over three decades. The majority of studies are related to linear systems but real-life systems are complex and nonlinear. The development of methodologies coping with complex and nonlinear systems have matured and even though there are many unsolved problems, methodology and associated tools have become available in the form of theory and software for design. Genuine industrial cases have also become available. Analysis of system topology, referred to as structural analysis, has proven to be unique and simple in use and a recent extension to *active structural* techniques have made fault isolation possible in a wide range of systems. Following residual generation using these topology-based methods, deterministic and statistical change detection has proven very useful for online prognosis and diagnosis. For complex systems, results from non-Gaussian detection theory have been employed with convincing results. The chapter presents the theoretical foundation for design methodologies that now appear as enabling technology for a new area of design of systems that are reliable in practise. Yet they are also affordable due to the use of fault-tolerant philosophies and tools that make engineering efforts minimal for their implementation. The chapter includes examples for an autonomous aircraft and a baling system for agriculture to illustrate the generic design procedures and real life results.

Keywords: Fault diagnosis, Fault-tolerant Control, Change Detection, Complex Systems

1 Introduction

Diagnosis of faults and active accommodation of faults are tools to prevent that faults develop into failure. Diagnosis is needed for fault-tolerant control where the diagnostic information is used without operator intervention to handle a fault or it is used by a human supervisor in support for fault-tolerant operation.

The theory of fault diagnosis has a long history. Early papers that helped the directions in this field included seminal results on generation of residuals using

parity space approaches [20], combined overview and research articles [40], [52], [40], [41], [55]. Early applications appeared in [79] for diagnosis of flight control systems, in [27] for chemical processes, in [54]. The theory for fault-diagnostic observers was pioneered in [32], [33], and generation of robust residuals in [35], [18]. Extension to fault-tolerant control emerged in [7], [9], [100] and theories for integration of diagnosis and control for linear systems appeared [73]. Diagnosis for nonlinear systems using geometric theory was pursued in [81]. Robust methods matured [59], [92] and [85], also for Fuzzy and neural network approaches [61], [78]. The crucial issue of threshold selection for diagnosis was treated in papers by [29] for the uncertain deterministic case and for the stochastic case by [3]. Extension to discrete event and quantised systems appeared in [70] and [69]. Early edited books helped develop the area [80], and several textbooks now show the maturity of the area [101], [2], [42], [72], [19], [47], [5], [71], [62], [56], [21], [76], [22].

The above methods rely mostly on detailed mathematical models of the plant. When complex systems are in focus, detailed models are very difficult or at least very expensive to obtain, and a different approach has become widely appreciated. This is based on graph-analysis of system properties and this is the subject of this chapter. The graph-based techniques do not replace the above analytical methods but for complex systems they have proven to be very attractive. Several aspects of control and diagnosis of complex systems appeared in [1]. Applications are plenty and to mention a few from recent years, omitting many important contributions: [8], [65], [46], [84], [93], [44], [13], [17].

Coping with complex systems is a challenge. Dimensionality of the problems met in real life is one challenge, complexity and nonlinearity are others. The ideas of using graph-analysis concepts to help solving complex set of equations were studied early in the applied mathematics community with the result of [25] being instrumental for the area. Further theoretical developments in [26] and [51] made the analysis of the structure of a set of equations or of a system described by such equations a feasible task, even for large systems of nonlinear equations. Structural analysis as this area is called, has been used intensively in Chemical Engineering for solving large sets of equations and issues on solvability have been pursued in a number of publications, see [96] and [68] and the references herein. The structural approach and the features it offers for analysing monitoring and diagnosis problems was first introduced in [89] and further developed in [88]. Extensions to analysis of reconfigurability and fault-tolerance emerged in [87] and [90]. The structural analysis approach was brought into a digested form in [5], [95]. Structural analysis has hence evolved during several decades. However, the salient features of the theory and the possibilities it offers have only become apparent to a larger community in the field of automation and automatic control over the last few years. Reasons for the slow penetration into applications origin mainly in the lack of widely available tools to support the structural analysis method for automated industrial systems. Software tools appeared in [23] and in [6]. An approach to highly efficient algorithms was developed in [63].

When considering diagnosis for control, the safety, from a structural point of view, depends on services offered by a system not only in normal operation but more important, after reconfiguration of the system to accommodate one or more faults. [11] considered the safety of fault-tolerant control schemes when multiple faults could be present. It was shown how structural analysis could be applied to analyse cases of multiple faults and to synthesise residual generators. Fault isolation, which is instrumental for correct fault handling, was addressed and active isolation was introduced from a structural point of view.

This chapter revisits the theory for graph-based analysis of systems, introduces the notion of structural active isolability. It first reviews the concept of behaviours and shows how the behaviour of a system is equally well applied on the services offered by hardware and software components. It then interprets the impact on safety of a system that is supposed to work under conditions of multiple faults. While it is well established that structural analysis is very useful for residual generation of technical processes [24], it is less obvious that the generic technique is also very applicable on complex system, even generation of residuals for diagnosis in a natural environment. The combination with change detection techniques is highlighted, and the techniques needed for nonlinear systems, when non-Gaussian residuals occur, are demonstrated. Two real cases are included for illustration of the complete design procedure, an aeroplane speed sensor fault diagnosis problem and a case of baler control for agriculture. The latter includes vision sensor techniques and results show how the diagnostic concepts can also be applied to enhance the robustness of a vision-based control system.

2 Graph-based Analysis

Fault-tolerant control uses control or sensor reconfiguration to accommodate failures in instruments, plant components or actuators. Aiming at utilising existing redundancy in instrumentation and control devices as far as possible, fault-tolerant control can be applied to minimise the hazards associated with malfunction, even when several sensors or actuators fail, but several modifications need be made to the usual single fault FTC schemes in order to achieve the necessary level of safety.

2.1 Reconfigurability and safety

The topology (structure) approach that is pursued in this context considers a system as consisting of a set of components which each offer a service and performs this service through defined normal behaviours. A component can offer different versions of services and command to the component can define which version of a service is made available. Within a component, fault-tolerant techniques can use fault-diagnosis and fault-handling to switch between services or offer a service in a version with degraded performance if local malfunction should make this necessary.

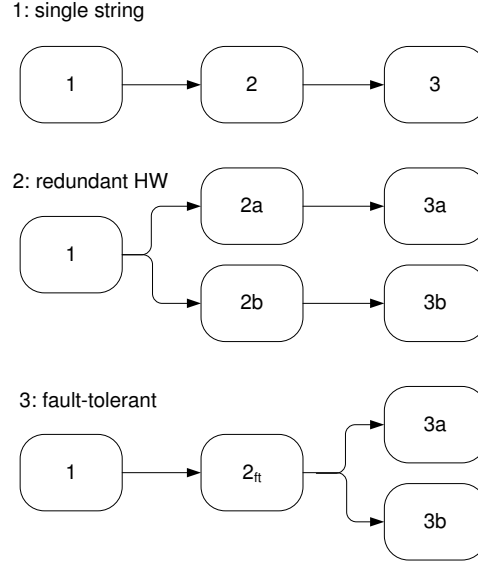


Fig. 1. Three architectures, single line with no redundancy (1), hardware redundancy (2) and combined fault-tolerance and redundancy (3)

2.2 Subsystem services

A system breakdown in Fig.1 shows three different topologies, by which we mean the arrangement of the system components and their interaction. Component k has input u_k , output y_k , parameters θ_k and a behaviour $c_k(y_k, u_k, \theta_k) = 0$. The behaviour may be constructed from a set of constraints $\{c_{k1}, c_{k2}, \dots, c_{kn}\}$ associated with the subsystem and the exterior behaviour of the component is the union of internal behaviours $c_k = c_{k1} \cup c_{k2} \cup \dots \cup c_{kn}$ or for brevity, $c_k = \{c_{k1}, c_{k1}, \dots, c_{kn}\}$. Following the generic component definition in [5], the service $S^{(k)}$ offered by component k is to deliver *produced variables* (output), based on *consumed variables* (input) and available *resources*, according to the specified behaviour $S_{(v)}^{(k)}$ where $v \in \{1, 2, 3, \dots\}$ is the version of the service. Clearly, the exterior behaviour is associated with the service offered by the component, we denote this behaviour by $c_k^{(v)}$.

In this context we particularly wish to consider versions of the same service that follow from the condition of the component, from normal over degraded to none. If a component has an internal failure, fault-tolerant techniques may still provide a version of the service with degraded performance $\left(S_{(d)}^{(k)}\right)$ or the service may not be available at all $\left(S_{(o)}^{(k)}\right)$. Hence, we consider the set of versions $v \in \{n, d_1, d_2, \dots, o\}$ where n : *normal*; d_1 : *degraded₁*; d_2 : *degraded₂*; o : *none*.

2.3 Service at system level

The service obtained by the system as an entirety is a function of the component architecture \mathcal{A} and the versions for the present condition k_i of components. With m components in a system, each component in one out of p conditions, $k_i \in \mathbb{N}^{p_i}$, we have a versions vector $\mathbf{v} = [v_1(k_1), v_2(k_2), \dots, v_m(k_m)]$, and the set of available behaviours $C_v = \{c_1^{v(k_1)}, c_2^{v(k_2)}, \dots, c_m^{v(k_m)}\}$.

Definition 1 (Overall Service) *The overall service available from a system is $S^{(s)}(c_i^{v(k_i)}) = \mathcal{A}(C_v|\mathbf{v}(\mathbf{k}))$, $i = 1, \dots, m$.*

With a single string architecture from Fig. 1, we obtain

$$S^{(s)} = S^{(1)} \cap S^{(2)} \cap S^{(3)} \quad (1)$$

With redundancy in the system, the hardware configuration with two parallel, totally redundant lines with only one component in common,

$$S^{(s)} = S^{(1)} \cap \left(\left(S^{(2a)} \cap S^{(3a)} \right) \cup \left(S^{(2b)} \cap S^{(3b)} \right) \right) \quad (2)$$

This solution is expensive as it requires two completely redundant subsystems. A cost effective solution would be to have some components intrinsically safe $S^{(1)}$, have others equipped with fault-tolerant properties so their service $S_{v(2)}^{(2)}$ will be available but in degraded version when local faults occur, and just have hardware redundancy for few essential components (3a, 3b). The fault tolerant architecture shown in part C of Fig. 1 is based on this idea. The service at system level is

$$S^{(s)} = S^{(1)} \cap S_{v(2)}^{(2)} \cap \left(S^{(3a)} \cup S^{(3b)} \right) \quad (3)$$

The paradigm in this architecture is that component failures should be detectable and control be switched to obtain a fault-tolerant service or reconfigure the system bypassing faulty components. This should be achieved by controlling the signal flow in the software of the system.

2.4 Availability and safety

The plant at the system level is *available* as long as the predefined normal service is offered in some version, normal or degraded. A fault-tolerant version of the service is obtained when one or more of the component services are offered in a fault-tolerant version. A fail-operational version of the service is obtained when hardware reconfiguration has been made to bypass a failure in the redundant component.

When multiple local failures are present, the service at system level is

$$S^{(s)} = \mathcal{A}(C_v|\mathbf{v})$$

Definition 2 (Availability) A system is available when $S^{(s)} \subseteq \mathcal{S}$ where $\mathcal{S} = \{S_1^{(s)}, S_2^{(s)}, \dots, S_n^{(s)}\}$ is the set of admissible services that meet specified overall objectives for behaviour \mathcal{O} of the system:
 $\forall S_i^{(s)}(c_i^{v(k_i)}) : C_v \subseteq \mathcal{O}$.

Definition 3 (Structural fault) Fault. A fault in the structural description of behaviours is a deviation from normal behaviour,
 $\exists i : c_i \neq 0$.

Definition 4 (Critical fault) A fault in c_i is critical $c_i \in C_{crit}$ if it will cause the system's behaviour to be outside the set of admissible behaviours,
 $c_i \in C_{crit}$ iff $c_i \neq 0 \Rightarrow C_v \not\subseteq \mathcal{O}$.

Definition 5 (Useability) Useability for reconfiguration. A faulty system is usable for reconfiguration, from the structure point of view, if all critical faults are structurally detectable,
 $\forall c_i \in C_{crit} : c_i \in C_{detectable}$

Assumption 1 (Intrinsical safety) It is a natural assumption that shut-down of the system is intrinsically safe and that the system can be shut down to a safe mode from any condition where $S^{(s)} \subseteq \mathcal{S}$.

Definition 6 (Structural reconfigurability) A system is structurally reconfigurable if
 $c_i \neq 0 \Rightarrow \exists j \neq i, \mathbf{v}(j) \neq \mathbf{v}(i) : C_{v(j)} \subseteq \mathcal{O}$

The task of fault-tolerant control is to find an appropriate $\mathbf{v}(j)$ when the fault c_i is detected and isolated and bring the system from version $v(i)$ to $v(j)$.

Having defined the system properties in terms of behaviours, it is natural to employ structural analysis where behaviours are defined in terms of constraints between variables and graph theory methods offer rapid and rigorous analysis.

2.5 Structure graph

A structural model of a system can be represented as a bipartite graph that connects constraints and variables. The structure graph [89] of a system (C, Z) is a bipartite graph $G = (C, Z, E)$ with two set of vertices whose set of edges $E \subseteq C \times Z$ is defined by $(c_i, z_i) \in E$ iff the variable z_i appears in constraint c_i .

The variables in Z are divided into known K and unknown variables X . Similarly, the constraints C are divided into constraints C_K that only apply to the known variables and C_X that involve at least one unknown variable. An incidence matrix S describes the structure graph where each row in the matrix represents a constraint and each column a variable. $S(i, j) = 1$ means that variable x_j appears in constraint c_i , $S(i, j) = x$ denotes a directed connection.

2.6 Constraints

Constraints represent the functional relations in the system, i.e. originating in a physical model using first principles. The constraints needed for structural analysis are far more simple. Instead of using the explicit system equations, structural analysis need to know whether a certain constraint makes use of a particular variable. Parameters that are known from the physics of the plant or from properties of the automation system, e.g. a control gain, are treated as part of the constraint in which the particular parameter is used. A constraint can be directed. This implies that a variable on the left hand side of the constraint can not be calculated from the right hand side of the constraint.

2.7 Variables

There are three different kinds of variables: *Input variables* are known, externally defined; *Measured variables* are entities measured in the system; *Unknown variables* are internal physical variables. Input and measured variables both belong to the set K but are separated for calculation of controllability.

2.8 Matching and results

The central idea in the structure graph approach is to match all unknown variables using available constraints and known variables, if possible. If successful, the matching will identify over-determined subgraphs that can be used as analytical redundancy relations in the system.

Use of a complete matching on unknown variables is not a necessary prerequisite to find analytic redundancy relations. Finding *minimal structural over-determined* subgraphs within a structure graph, referred to as MSO sets, was introduced in [64], and their very efficient MSO set algorithm makes structural analysis feasible for real life complex systems.

Results of the structural analysis are

- List of parity relations that exist
- Auto-generated suggestion of residual generators
- List of detectable behavioural faults
- List of isolable behavioural faults

The term *behavioural faults* are used to emphasise that the faults determined by the structural analysis are violation of a normal behaviour.

When a matching has been found, backtracking to known variables will give a suggestion for parity relations that could be used as residual generators. A system with m constraints and n parity relations will give a relation showing which residuals depend on which constraints.

One view on these relations is the boolean mapping,

$$\mathcal{F} : r \leftarrow M \otimes (c_i \neq 0) \tag{4}$$

from which structural detectability and isolability can be found.

Definition 7 (Structural detectability) A fault is structurally detectable iff it has a nonzero boolean signature in the residual,
 $c_i \in C_{detectable}$ iff $\exists j : c_i \neq 0 \Rightarrow r_j \neq 0$

Definition 8 (Structural isolability) A fault is structurally isolable iff it has a unique signature in the residual vector, i.e. column m_i of M is independent of all other columns in M ,
 $c_i \in C_{isolable}$ iff $\forall j \neq i : m_i \neq m_j$

2.9 Active isolation

In some cases faults are group-wise isolable, i.e within the group individual faults are detectable but not isolable. This implies that with the given architecture of the system, these faults are group-wise not isolable. This does not necessarily imply that isolation can not be achieved in other ways. Indeed, although the same set of residuals will be "fired" when either one or the other of non-structurally isolable constraints is faulty, the time response of the residuals may be different under the different fault cases. Exciting the system with an input signal perturbation may therefore make it possible to discriminate different responses of the same residual set when different constraints within the group are faulty. The analytical idea of applying test signals to isolate faults is not new. [101] designed test signals for diagnosis. A sophisticated setup for active diagnosis was made in [74] and controller switching made for active diagnosis was shown in [83]. The structural analysis approach was first suggested in [11] and extended to different use modes in [67].

Proposition 1 Active structural isolation is possible if and only if both a structural condition and a quantitative condition are true.

Structural condition : the known variables in the set of residuals associated with a group of non-structurally isolable constraints include at least one control input. Quantitative condition : the transfer from control inputs to residuals and or to output is affected differently by faults on different constraints.

Definition 9 (Input to output reachability) Let $p^{(i,j)} = \{c_f, c_g, \dots, c_h\}$ be a path through the structure graph from input u_i to z_j , where z is a residual or an output and $\prod^{(i,j)}$ the union of valid paths from u_i to z_j . Let $C_{reach}^{(i,j)} = \{c_g \mid c_g \in \prod^{(i,j)}\}$. A constraint c_h is input reachable from input u_i if a path exists from u_i to any output (or residual) z_k and the path includes the constraint, $c_h \in C_{reach}^{(i,k)}$

Proposition 2 (Active structural isolability) Two constraints c_g and c_h are actively structurally isolable from output signatures if $\exists i, j, k, l : c_g \in C_{reach}^{(i,j)}, c_h \in C_{reach}^{(k,l)}$ and $\{c_g, c_h\} \notin C_{reach}^{(i,j)} \cap C_{reach}^{(k,l)}$

These structural properties easily reveal which possibilities there are for active isolation of faults in a system described by its behaviours and associated

topology (structure). The detailed design of which test signals are feasible and how test signals are detected efficiently are subjects of the signal-based design that follows the analysis based on structure.

2.10 Analysis of scenarios with multiple faults

Scenarios of multiple faults are dealt with, in the structural analysis context, by removing one or more constraints that represent the faulty parts of the system. Should c_6 be subject to a local failure, the remaining system $\mathbf{S}_f = \mathbf{S} \setminus \{c_6\}$ need be re-analysed. The results can show which residual generators exist for the faulty system, and which further faults could be isolated or detected. An application to a marine control system was treated in [8] where analysis of multiple faults was demonstrated as part of a fault-tolerant design.

3 Design procedure based on analysis of behavioural relations

Having introduced the formal background of the structure-based approach, the generic procedure for setup and design of the diagnostic system is now presented, including a brief overview of the signal-based design steps that are needed to convert the results from analysis based on structure to detection filters and change detection algorithms. This section will outline the generic design procedure [5] based on a system described by dynamic and algebraic constraints,

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{g}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{d}, \theta) \\ \mathbf{0} &= \mathbf{m}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{d}, \theta) \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{d}, \theta) \\ \dot{\mathbf{x}} &= \frac{d\mathbf{x}}{dt},\end{aligned}\tag{5}$$

where \mathbf{x} are states, \mathbf{z} are variables determined by algebraic constraints, \mathbf{u} is input, \mathbf{y} is measured output, \mathbf{d} is unknown disturbances and θ are system parameters.

3.1 Tools for analysis of system structure properties

Let a system be given by the set of constraints of the form 5. The basic tool for analysis of system structure and generation of analytical redundancy relations is the matching of the unknown variables $(\mathbf{x}, \mathbf{z}, \mathbf{d})$ in a bipartite graph to the known ones, (\mathbf{y}, \mathbf{u}) through the constraints, as outlined above.

A first approach to matching in structural analysis was to generate a single complete causal matching on the over-specified part of the system [89] from which structural detectability and isolability properties could be shown. For nonlinear systems, isolability can be enhanced by the use of combinations of residuals for a diagnosis. Therefore, all matchings were generated in [23, 28] to investigate isolability properties. Finding all possible matchings is computationally heavy or

even impossible for large industrial scale systems and Krysanter [63], [64] instead finds minimal structurally over-determined (MSO) subgraphs within a graph. An MSO set is a subgraph where only one constraint can be removed to get a just-determined part of the graph. An MSO set contains a complete matching plus one unmatched constraint, and one redundancy relation is available from each MSO set. The algorithm by [63] is very efficient and was found to be about 10^5 times faster for a nine-bus electrical power distribution system [60] than the earlier algorithms [28]. In cases where the system topology might change, further techniques exist to also explore the different modes of operation and deduct isolability from the operation in different *use modes* [67].

3.2 Transformation to signal space and analysis

Having found the matchings or MSO sets from the structural representation, and having listed the from the unmatched constraints to be used for diagnosis, mapping to analytical form is obtained by *backtracking* of unknown variables to known ones. Solving for unknown variables is done in the order specified by the particular matching. Tools are available that can compute the residuals from specification of the constraints [6], from a Simulink[®] simulation of a system [38] or from a bond graph of the system, see [30] and references herein.

FDI properties from the structural analysis should be interpreted with a bit of care, since structural and analytical properties in detectability and isolability are not identical. The isolability relations were analysed in [66] who introduced *checking models* to better approximate analytical isolability from structural isolability.

In summary, analytical redundancy relations (ARR) are generated from unmatched constraints, and the residuals are the ARR functions of time. The general form of the analytical redundancy relation obtained is the vector $r(t)$, which is *zero* or *non-zero* when there is no deviation from normal, the \mathcal{H}_0 condition, or a deviation from zero, the \mathcal{H}_1 condition, respectively.

The set of constraints \mathbf{c}_{arr} that form the ARRs obtained from structural analysis will be linear or nonlinear, according to how the system is modelled,

$$\mathbf{c}_{arr}(t) = \mathbf{g}_{arr}(y, u, x, \dot{x}, \theta), \quad (6)$$

where θ are system parameters. Backtracking through a matching will express the unknown variables by the known ones, hence the set of residuals, represented by the vector $\mathbf{r}(t)$ will be functions only of known variables and their derivatives,

$$\mathbf{r}(t) = \mathbf{g}_r(\mathbf{y}(t), \mathbf{u}(t), \dot{\mathbf{y}}(t), \dot{\mathbf{u}}(t), \ddot{\mathbf{y}}(t), \ddot{\mathbf{u}}(t), \dots, \theta). \quad (7)$$

Residuals for diagnosis In terms of structural analysis, $\mathbf{r}(t)$ will be different from zero for some t if a constraint is violated. In order to analyse the detailed properties of residual generators, we introduce faults \mathbf{f} in an explicit way in the system being analysed and can then deduct properties as sensitivity of residuals

to particular faults and estimation of magnitude of faults. The system is then described by

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{g}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{d}, \theta, \mathbf{f}) \\ \mathbf{0} &= \mathbf{m}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{d}, \theta, \mathbf{f}) \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{d}, \theta, \mathbf{f}) \\ \dot{\mathbf{x}} &= \frac{d\mathbf{x}}{dt},\end{aligned}\tag{8}$$

and the associated residual vector by

$$\mathbf{r}(t) = \mathbf{g}_r \left(\mathbf{y}(t), \dot{\mathbf{y}}(t), \dots, \mathbf{u}(t), \dot{\mathbf{u}}(t), \dots, \mathbf{f}(t), \dot{\mathbf{f}}(t), \dots, \theta \right). \tag{9}$$

In the linear, time invariant setting, using s as the Laplace transform variable, the residual vector can be written in the form,

$$\mathbf{r}(s) = \mathbf{V}_{ru}\mathbf{u}(s) + \mathbf{V}_{ry}(s) (\mathbf{H}_{yu}(s, \theta)\mathbf{u}(s) + \mathbf{H}_{yd}(s, \theta)\mathbf{d}(s) + \mathbf{H}_{yf}s, \theta\mathbf{f}(s)) \tag{10}$$

Decoupling from input \mathbf{u} and disturbances \mathbf{d} are obtained as part of the structural analysis since unknown disturbances are matched. Hence \mathbf{V}_{ru} and \mathbf{V}_{ry} obtain the property,

$$\mathbf{0} = (\mathbf{V}_{ry}(s) \mathbf{V}_{ru}(s)) \begin{pmatrix} \mathbf{H}_{yu}(s) & \mathbf{H}_{yd}(s) \\ \mathbf{I} & \mathbf{0} \end{pmatrix} \tag{11}$$

This could also be obtained through a left nullspace design in the frequency domain. For linear systems, the decoupling property could also be obtained by unknown input observer design [34]. The structural analysis, the left nullspace approach and the unknown input observer design are equivalent for stable systems. Structural analysis followed by an observer-based implementation is preferred for unstable systems.

Real life residuals should not comprise pure derivatives, for reasons of noise amplification, and no pure integrations, for reason of bias in measurements, so filtering is required. This is a natural part of the design that follows the basic calculation of residuals. Linear methods have been thoroughly analysed in [97] and references herein.

Ideal residual generators, leave the residual perfectly decoupled from input and disturbances. The residuals are only sensitive to faults, but the decoupling requirement can sometimes mask particular faults. When a non-masked fault is present, the residual will change, and detection of the change makes room for diagnosis of the fault. The logical steps include first *detection* that a fault is present, then *isolation* of which fault is present and then *estimation* of the magnitude of the fault. Faults can be *strongly detectable* in residuals if the change in residual persists as long as the fault is present, or it can be *weakly detectable*. The definitions of these terms are provided in the standard literature on fault diagnosis, including text books as [80],[42], [5] and [21].

3.3 Robust residuals

The ideal performance of residual generators Eqs. (9) and (10) is obtained assuming perfect knowledge of system dynamics. With uncertainty being present, robustness need be assured. The structural analysis approach can force residual generators to be robust to selected parameters by declaring these parameters as unknown variables in the analysis. In this way, the parameters are treated as completely uncertain. If robustness is desired against more generic uncertainty bounds, not related to one or more particular parameters, principles of robust design using \mathcal{H}_{∞} optimal design methods were thoroughly treated in [72] and [19]. The principle is to define residual generation as an optimisation problem, weighing sensitivity for faults against sensitivity to input and to unknown disturbances. Robustness to variations and uncertainty in parameters θ can be obtained using \mathcal{H}_{∞} methods. The robust residual generator design has been extended to incorporate active diagnosis techniques [74].

Another approach is to use interval methods where an observer calculates the interval, a polygon in the output space, where output could be expected given the bounds on the θ vector. A prediction can also be made assuming the occurrence of particular faults and associated intervals [53]. Hypothesis about faults can be made by rejection of models that are not in conformance with observed behaviour [16],[86], and robust active diagnosis methods have been suggested [94].

3.4 Evaluation of residuals

In most real applications, random noise in residuals requires that stochastic change detection is employed for their evaluation and testing of hypotheses about presence of faults. Using the notation \mathcal{H}_0 for nominal system (no faults) and \mathcal{H}_1 for occurrence of a particular fault. Two models are distinguished based on discrete time observation of the residual,

$$\begin{aligned}\mathcal{H}_0 : \mathbf{r}(k) &= \mathbf{w}(k) \\ \mathcal{H}_1 : \mathbf{r}(k) &= \mathbf{a}(k) + \mathbf{w}(k)\end{aligned}\tag{12}$$

where $\mathbf{a}(k)$ is a change in the residual of known or unknown magnitude or time-wise profile, and $\mathbf{w}(k)$ is a random component that can have a Gaussian or other distribution. The increments of each component $w_i(k)$ of $\mathbf{w}(k)$ would ideally be independent and from identical distributions (IID). Change detection is often based on testing a log-likelihood ratio of the distribution of residuals $p(r|\mathcal{H}_0)$, assuming \mathcal{H}_0 and $p(r|\mathcal{H}_1)$, assuming \mathcal{H}_1 . Taken over a window of size M , the log-likelihood

$$S(k) = \frac{1}{M} \sum_{j=k-M}^k \ln \left(\frac{p(r(j)|\mathcal{H}_0)}{p(r(j)|\mathcal{H}_1)} \right).\tag{13}$$

Theoretical methods to detect various change profiles have been developed, see [2] and [75] for Gaussian change detection, [58] for non-Gaussian cases and [47]

for combined signal processing and change detection, and improved formulations are constantly evolving, see e.g. [31].

A test statistic $g(k)$ is computed based on $S(k)$ using cumulative sum (CUSUM) algorithms for known changes, generalised likelihood (GLR) methods or others to detect an unknown change. A popular detection scheme is to compare $g(k)$ with a threshold h and decide \mathcal{H}_1 when $g(k) > h$. Key features offered by change detection theory include the possibility to predict the probability of false detection P_F under \mathcal{H}_0 and the probability of detection P_D under \mathcal{H}_1 . Theoretical results to determine h commonly assume IID to calculate a detection threshold

Real life residuals often are neither Gaussian nor IID so theoretic thresholds fall short in relating practical choice of h with P_F and P_D . Instead, the distribution of $g(k)$ can be estimated from data, and h is determined from the estimated distribution. Estimating over a sliding window was pursued in [39] and [48], [99] for different domains of applications.

4 Aeroplane Diagnosis and Fault Handling

A case of diagnosis and fault handling of speed sensor faults on a small Unmanned Aerial Vehicle (UAV) was first studied in [49] and a comprehensive solution was presented in [48]. Focusing on faults in the pitot tube, that easily causes a crash if not diagnosed and handled in time, redundant information about aircraft velocity, residuals are easily generated that allow both isolation and handling of a single sensor failure. Recorded telemetry data of an actual event with a pitot tube defect illustrate the efficacy of the diagnostics.

4.1 Airspeed Sensor Problem

Defects on sensors can have catastrophic consequences for aircraft, specially smaller UAVs, which do not have the same sensor redundancy that is available on a larger aircraft. It is therefore important to be able to detect whether a sensor defect has occurred. One of the vital sensors for an UAV is the pitot tube which measures the airspeed of the vehicle. This sensor is very exposed because of its position in the airstream and can easily be clogged by dust or water particles that freeze at higher altitudes.

The solution to these clogging problems usually employed on larger aircrafts is to install several pitot tubes with build in heating devices to have a redundant system that can accommodate icing. Because of weight and space limitations, adding more sensors is usually not an option on smaller UAVs. Therefore, a different approach must be taken to diagnose and accommodate faults. One way is to have artifact readings detected and replaced with estimated values. Detection of faults and fault-tolerance for UAVs has a lot of focus and, as described in [22], many parts of the aircraft control and operation can benefit from using fault tolerant methods. A systematic approach to fault detection is described in [36] and some of the applications of these methods are, detection of mechanical defects, like stuck control surfaces. These were studied in [4] and [77] where

active methods were used to isolate faults. Observer based fault diagnosis was investigated in e.g. [50]. Nonlinear models that describe the aircraft can also be used in fault diagnosis, this was demonstrated on small helicopters in [37]. A comprehensive overview for the state of art in diagnosis in aerospace was given in [103], and [46] and [43] covered uses in the aircraft industry.

4.2 Model for Diagnosis

Aircraft are usually modelled by the dynamic and kinematic equations, that describe their motion through the air. For some aircraft very detailed models are available, that includes nonlinear terms caused by aerodynamic effects. As the equations of motion are inherently nonlinear, these models gives a very accurate aircraft behavioral description. Several textbooks, including [91] describe the detailed models.

A formulation of the generic behaviours for an aircraft [36], showed that the structural analysis approach is also well suited for generating a model for fault diagnosis. The behavioural formulation reduces the complexity to precisely what is required for diagnosis and the details of particular nonlinearities need be scrutinised only when needed for particular residuals. Focusing on the speed sensor fault diagnosis, a simplified behaviour description, related only to vehicle speed, is shown in Table 1 where \mathbf{v}_n is speed over ground (navigation frame), v_t is air speed through propeller disc if the propeller was not present, \mathbf{v}_a is air speed vector, \mathbf{v}_w , the wind velocity vector, v_p pitot tube measured air speed, \mathbf{v}_g velocity estimated by the GPS.

Table 1. Velocity Related Behaviours

constraint	behaviour
c_1	$: \mathbf{v}_n = \mathbf{v}_a + \mathbf{v}_w$
c_2	$: \hat{v}_t = g_1(n, \mathbf{v}_a , \theta)$
c_3	$: \hat{v}_t = \mathbf{v}_a $
c_4	$: \hat{\mathbf{v}}_w = g_2(\mathbf{v}_n, \mathbf{v}_a)$
c_5	$: \hat{\mathbf{v}}_w = \mathbf{v}_w$
m_1	$: \mathbf{v}_g = \mathbf{v}_n$
m_2	$: v_p = \mathbf{v}_a $

In Table 1, the function g_1 is an estimator of air speed \mathbf{v}_a using the propeller thrust relations, where n is known rotational speed, g is gravity acceleration constant, m is mass of the aircraft, T_{nn} and T_{nv} are parameters determined from propeller characteristics, and F_A is aerodynamic drag,

$$g_1 = (T_{nv}n)^{-1} (T_{nn}n^2 - F_A(v_a, \theta) - mg \sin(\theta)) \quad (14)$$

The function g_2 is an estimator that provides a fairly uncertain estimate of the wind velocity vector.

Table 2. Graph for Aeroplane Velocity Case.

	known				unknown				
	n	\mathbf{v}_g	v_p	θ	\hat{v}_t	\hat{v}_w	\mathbf{v}_n	\mathbf{v}_a	\mathbf{v}_w
c_1							1	1	1
c_2	-1				1			-1	
c_3			-1		1			-1	
c_4						1	-1	-1	
c_5						1			1
m_1		1					1		
m_2			1					-1	

The sets \mathcal{K} and \mathcal{X} of known and unknown variables in this problem are,

$$\begin{aligned}\mathcal{K} &= \{n, \mathbf{v}_g, v_p, \theta\} \\ \mathcal{X} &= \{\hat{v}_t, \hat{v}_w, \mathbf{v}_n, \mathbf{v}_a, \mathbf{v}_w\}\end{aligned}\tag{15}$$

With 5 unknowns and 7 constraints, there are up to 2 unmatched constraints, that can be used as parity relations. When solving the set of matched constraints in Table 1, calculability need be accounted for since a vector length can be determined from the components of the vector but not reversely. A representation of the structure graph associated with the constraints are shown in form of the incidence matrix in Table 2. A 1 in the matrix shows that the associated variable can be calculated from the constraint, a -1 shows that it can not.

One complete matching on the unknown variables is achieved as follows:

$$c_1 \rightarrow \mathbf{v}_a; c_4 \rightarrow \hat{\mathbf{v}}_w; c_5 \rightarrow \mathbf{v}_w; m_1 \rightarrow \mathbf{v}_n; c_3 \rightarrow \hat{\mathbf{v}}_t;\tag{16}$$

The set of unmatched constraints are $\{c_2, m_2\}$. Backtracking to known variables along the matching leads to a symbolic form of the residuals and insertion of the analytical constraints from Table 1 gives the residuals,

$$R_1 = |\mathbf{v}_g - \hat{\mathbf{v}}_w| - v_p = |\mathbf{v}_g - g_2(\mathbf{v}_n, \mathbf{v}_a)| - v_p\tag{17}$$

$$R_2 = \hat{v}_t - v_p = g_1(n, |v_p|, \theta) - v_p\tag{18}$$

This result is quite intuitive but the formal procedure assures that all possible redundancies have been explored.

Finding all minimal structurally overdetermined subsystems in a set of constraints is the basis of an extremely efficient algorithm developed by Krysander et.al [64]. The result is a table similar to the backtracking shown above that is a recipe to find an analytic redundancy relation. This algorithm is integrated in the latest version of SaTool, an open source software [6] that analyze detectability and isolability, and provides symbolic and analytic expressions of residuals from a set of constraints that express the dynamic behaviours of a system.

While the simple estimate g_1 was found to suffice for the purpose of fault diagnosis by [49], more sophisticated and precise nonlinear estimators have been

suggested in the literature. [102] described a way to estimate state and parameters in nonlinear systems with a structure similar to the thrust equation here, [10] applied an adaptive observer scheme and [82] showed nonlinear observer designs for thrust estimation. Several estimation techniques were integrated in determining air-speed sensor faults in [48] that also assessed detection and false alarm probabilities from real data.

4.3 Signal analysis

Figure 2 shows a time history and a histogram for R_1 and R_2 from UAV flight data in the fault free case.

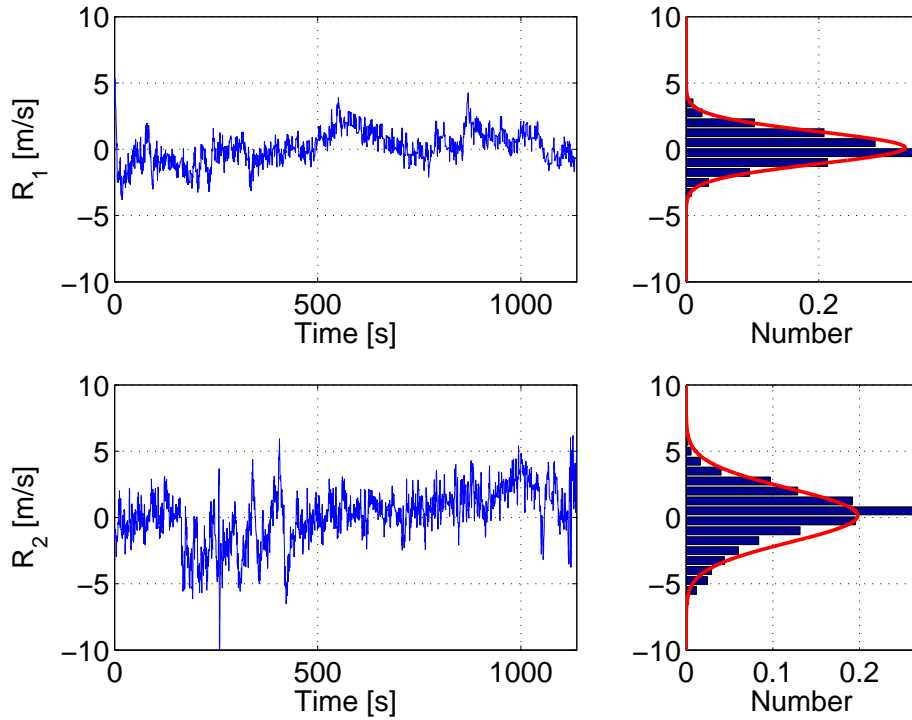


Fig. 2. Time development and histogram for residual R_1 and R_2 in the fault free case.

The power spectral densities of the residual signals are not white. Since white noise is one of the requirements for most statistical change detectors to perform optimally, the low-pass filtered nature of the noise should be removed. One solution of dealing with coloured noise is to filter white noise through a suitable filter function to take into account the correlations present in the coloured noise. This can be created from a large record of data where all the signals properties are

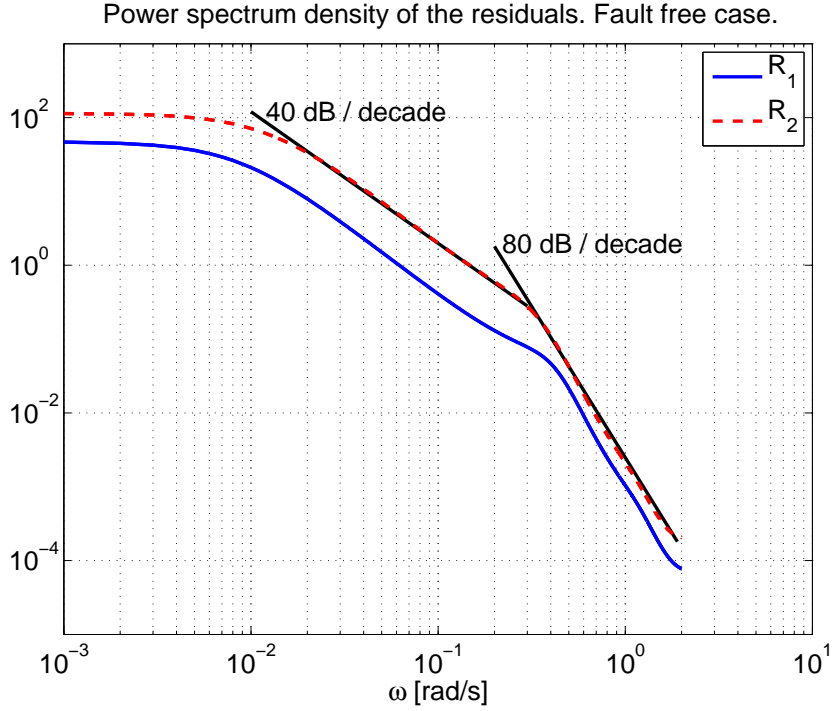


Fig. 3. Power spectrum densities for residual R_1 and R_2 .

present. As indicated on figure 3 the power spectrum density of the two residuals consists of a part which decreases with 40 dB/decade and a part which decreases with 80 dB/decade. A whitening filter can be implemented as any stable filter, including as a Kalman filter.

Figure 4 shows time series of the whitened residuals and their histograms. Neither of the residuals are now Gaussian distributed. Instead they follow the Cauchy distribution (equation 20) very well as indicated by the green dotted line in the histogram. The Gaussian nature that the residuals had before whitening apparently came from the effects of low-pass filtering. When removed during the whitening process, the Gaussianity was lost.

4.4 Change Detection

Detection of an unknown change of mean value A of a signal x should distinguish between two hypotheses,

$$\begin{aligned} \mathcal{H}_0 : x[n] &= w[n] \\ \mathcal{H}_1 : x[n] &= A + w[n] \end{aligned} \quad (19)$$

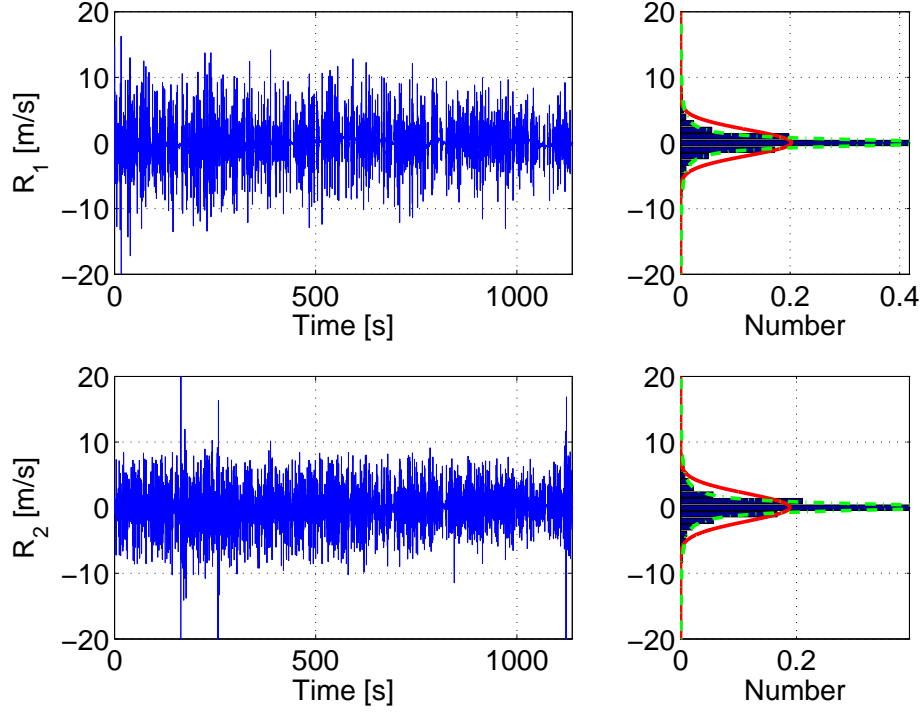


Fig. 4. Time development and histogram for the whitened residual R_1 and R_2 in the fault free case.

The variance of the noise w is σ_w^2 . The GLRT (Generalised Likelihood Ratio Test) is a standard way to solve such a problem. The standard solution assumed Gaussian noise. When the residuals are distributed according to a Cauchy distribution (as seen from figure 4) the following probability distribution function must be used

$$p(x; x_o, \beta) = \frac{\beta}{\pi (x - x_o)^2 + \beta^2} \quad (20)$$

where the two parameters are the half-width half-maximum scaling, β , and the offset x_o . The GLRT test statistic becomes

$$L_G(\mathbf{x}) = \frac{\prod_{i=1}^N p(x_i; \hat{x}_o, \hat{\beta})}{\prod_{i=1}^N p(x_i; 0, \hat{\beta})} > \gamma_c \quad (21)$$

The maximum likelihood estimate (MLE) of $\hat{\beta}$ and \hat{x}_o is found by fitting the data to equation 20.

Detection of change according to Eq. 21 requires that the threshold γ is determined. If the distribution of $L_G(\mathbf{x}|\mathcal{H}_o)$ is known, the probability of false

alarm is:

$$P_F(\mathbf{x}|\mathcal{H}_0, \gamma) = \int_{\gamma}^{\infty} p_{L_G}(x|\mathcal{H}_0)dx \quad (22)$$

The probability density (pdf) and the cumulative density (cdf) for $L_G(\mathcal{H}_0)$ are shown using flight data in Fig. 5.

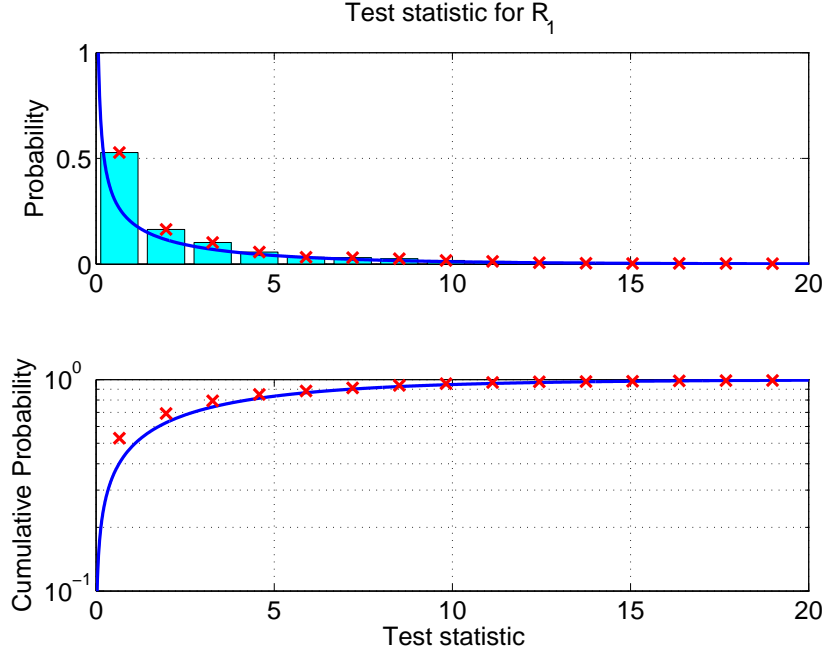


Fig. 5. Histogram and cumulative density for the $L_G(\mathcal{H}_0)$ test statistics for the GLRT Cauchy detector. Data and approximating function.

The approximating function to the CDF in the figure is a Gamma distribution.

$$G(x; a, b) = \frac{1}{b^a \Gamma(a)} x^{a-1} \exp\left(-\frac{x}{b}\right) \quad x > 0 \quad (23)$$

where Γ denotes the Gamma function. The distribution is fitted with a MLE and the following parameters are found

	a	b
R_1	0.46	5.58
R_2	0.43	4.91

Selecting a threshold of $\gamma = 50$ gives a probability of false alarm of $1 - G(50; 0.46, 5.58) \cong 2.0 \cdot 10^{-5}$ for R_1 .

The theoretical performance could also be calculated using formulas from [49] but since theory and practice sometimes differ, it is advisable to check one's threshold value using an observed CDF from the test statistics of real data. This is particularly the case when the assumption of independent and identically distributed (*iid*) samples is not valid. The threshold required to achieve a certain false alarm probability can be very much different from its theoretical value [58] when the *iid* assumption is not valid.

4.5 Results

To test the detection performance, data from a real event are used where a UAV crashed, caused by a pitot tube defect and propagated effects on the flight control. The detection results are shown together with the residuals in Figure 6.

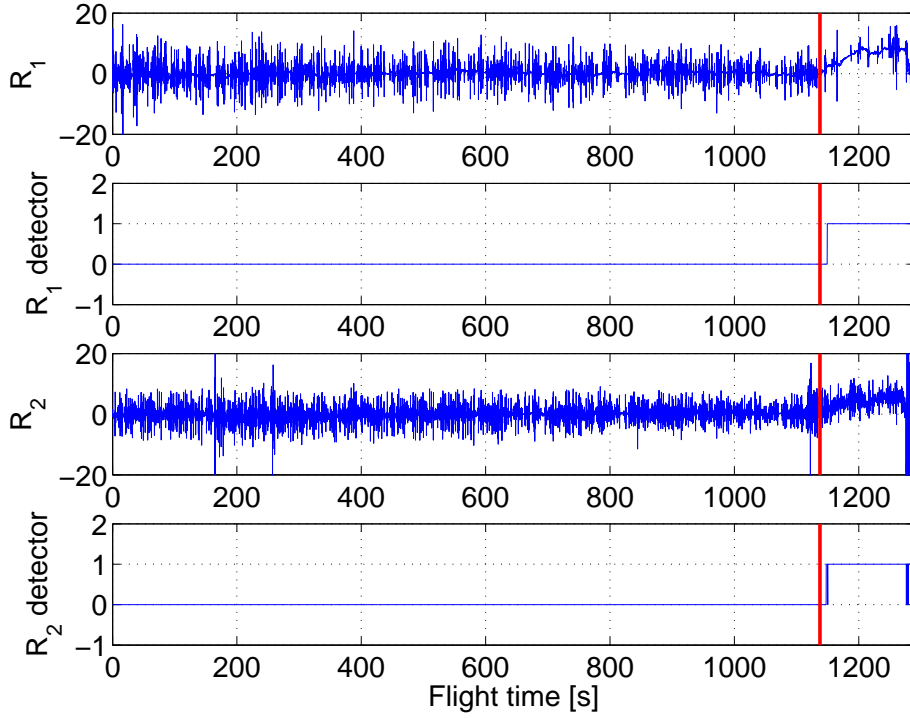


Fig. 6. Residuals and detector output. At approximately $t = 1140$ s the clogging of the pitot tube occurs (marked with the red vertical line).

As seen an alarm is raised, indicated by a value of 1, about 14 seconds after assumed instant when the incident started. Timely detection is hence obtained

and with yet another 120 seconds elapsing until the crash happened, there would have been ample time to execute remedial actions had this detection system been available.

The simple detection methods in this example are sufficient to avoid simple accidents with simple equipped UAV's. Pitot tube defects have happened also on larger aircraft despite redundancy to a fail-operational level and the fault-diagnosis techniques could be part of fault-tolerant solutions and have a wider area of application.

5 Fault-tolerant Guidance using Vision

This second case from agriculture brings diagnosis and fault-tolerant control techniques in operation within advanced computer vision based sensors. In agriculture, typical tasks are to follow structures in the field to plow, seed, spray or harvest. The specific harvesting task of baling involves to follow rows of cut straw or grass (swath, see fig. 7) in order to pick it up and process it into bales. This is a labour intensive and repetitive task, which is of interest to automate. The difficulties pertaining to automating this task are similar to the difficulties in automating a large range of agricultural tasks. The ability to track this structure using 3D shape information from a stereo camera and/or GPS information was demonstrated in [15] and a detailed presentation of the baling problem was presented in [13] where a classifier was employed based on online learning of texture information about the swath and the surroundings. This was then coupled with shape information to extract the swath position and a mapping kept track of measured swath positions. The map was used to guide the vehicle along the swath by steering the tractor's front wheels while a driver controlled the throttle and brakes. The novelty with respect to diagnosis in this application is the use of diagnostic techniques to obtain fault tolerance in the stereo vision sensor itself, avoiding typical reasons for artifacts that occur in the stereo vision process when distance to objects are calculated. Highlights from this real-life case are given below.

The system to be analysed is equipped with stereo-vision, a single antenna GPS and an IMU, in one configuration. GPS positions of the vehicle that formed the swath are known. The combination of stereo-vision and GPS allows the system both to "see" the swath but also navigate based on the given map. This creates system redundancy that is essential for achieving fault-tolerance. A visual odometry algorithm on the stereo-camera allows for the relative position of the vehicle without GPS or IMU. The GPS receiver used was ground-station compensated and the IMU was a tactical grade (low accuracy) MEMS based unit.

The two main ideas presented is first a behavioural model for representing the natural environment, namely the swath. Secondly, it is shown how parts of this model (the swath location) can be used in conjunction with sensor inputs to create a fault tolerant sensor fusion system. The fault diagnosis is illustrated using real data.



Fig. 7. Swath in the field to be collected.

5.1 Modelling the Natural Environment

A model of the swath requires extracting the salient features of the environment required for the field operation and storing them in the model representation. The salient features are the location of the swath and the distribution of the swath material across the swath. Swath location, width, and height are modelled as a map using splines to follow the centre of the swath in a metric map. The model is illustrated in fig. 8.

The swath location is defined as being in a 2D coordinate system on the ground plane. A function f represents the lines down the middle of the swaths. Given coordinate pairs (x, y) then f is:

$$y = f(x) \quad (24)$$

The model of the swath location is then $s(x)$ with $s \in \mathcal{S}_3(\mathbf{k}_{0:n})$, where $\mathbf{k}_{0:n}$ are the spline knots and the spline coefficients and \mathcal{S}_3 is the cubic spline domain. Then the model is equal to the swath location plus the approximation error e_a of fitting a spline to f :

$$s(x) = f(x) + e_a \quad (25)$$

Based upon the concept of having a controller that allows the vehicle to follow the swath location, the position error $\varepsilon_x = [y_0, \phi, C_0]^T$ can be found as a function of the tractor position and the spline. Here, y_0 is the lateral offset from the control point to the spline (also known as the X-track error), angle

ϕ is the angular difference between the vehicle attitude and the swath, and C_0 is the curvature of the swath near the vehicle. Defining the spline s_b in body coordinates a function \mathcal{E} can be set to find the position error:

$$\varepsilon_x = \mathcal{E}(s_b) \quad (26)$$

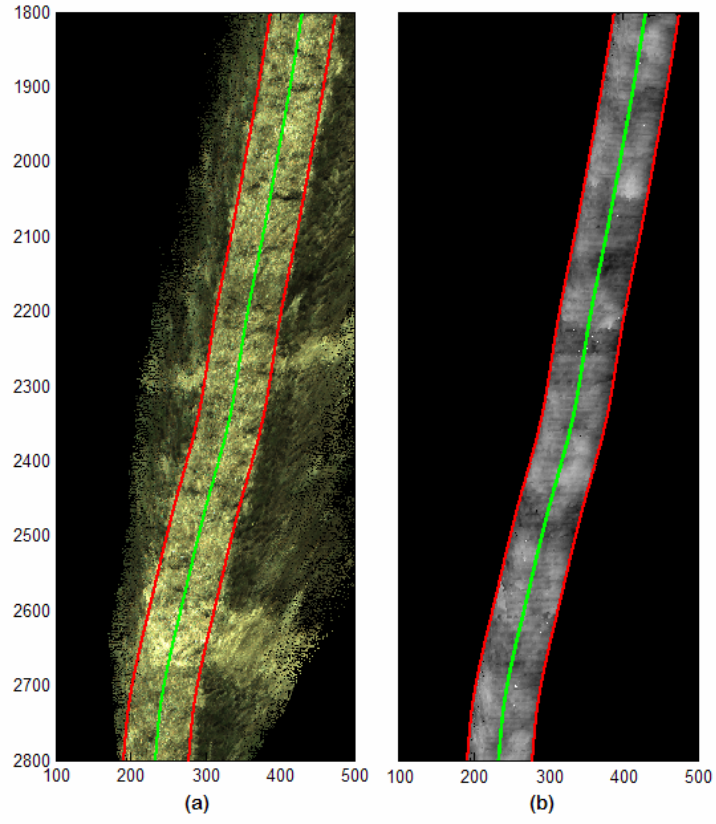


Fig. 8. (a) RGB topdown-view of the swath with swath location illustrated with a green spline and estimated swath width superimposed with red borders. (b) The same swath model but with grayscale values illustrating the height of the swath in a given cell.

5.2 Stereo-Camera

A stereo algorithm is used to find the correspondence between features in the left and right image sensor (i_l, i_r) . The position of the features relative to the stereo-camera can then be inferred in 3D. Modern vision algorithms then exist to register 3D models with the 3D point cloud provided by the stereo-camera: [45]. An algorithm has been constructed that allows such registration between the swath model and the 3D points. The stereo-algorithm and registration will be denoted by the function \mathcal{V}_{reg} . Thus, given the two images a measurement of the swath location s_c can be computed for the part of the swath in the image.

$$s_c = \mathcal{V}_{reg}(i_l, i_r) \quad (27)$$

These measurements are stored in a map representation for an individual swath and s_m is the spline formed by combining N previous measurements,

$$\mathbf{s}_m(k) = \min_{spline} (s_c(j), j = k - N, k) \quad (28)$$

5.3 Robust Stereo Enhancement by Texture

Stereo vision uses correlation of textures to determine distance to objects in view of both cameras. When misclassification of texture happens, gross errors may occur in the stereo calculated distance to objects or artifact objects can occur or objects can disappear. To robustify the stereo against such artifacts, [13] introduced statistical validation of texture using so called *textons*. Texture properties can include colour distribution, intensity, shape and patterns. Textons are basis vectors extracted from the local descriptors of texture. Following [98], who showed that small local texture neighbourhoods may be better than using large filter banks, [14] and [13] employed statistical classification to the problem and used the texture-based classification of swath as a supplement to the geometrical mapping described above.

5.4 Texton Labelling

Given a colour image as input, pixel neighbourhoods in the image are grouped into belonging to a texton type, which belongs to a set of basis textons v_i obtained by prior learning from a training image. This is done by first extracting a descriptor in the form of a vector from each pixel location in the image. For each location the p_i vector is:

$$p_i = \begin{bmatrix} w_1 * l_c \\ w_2 * a_c \\ w_2 * b_c \\ w_3 * (l_1 - l_c) \\ \vdots \\ w_3 * (l_8 - l_c), \end{bmatrix} \quad (29)$$

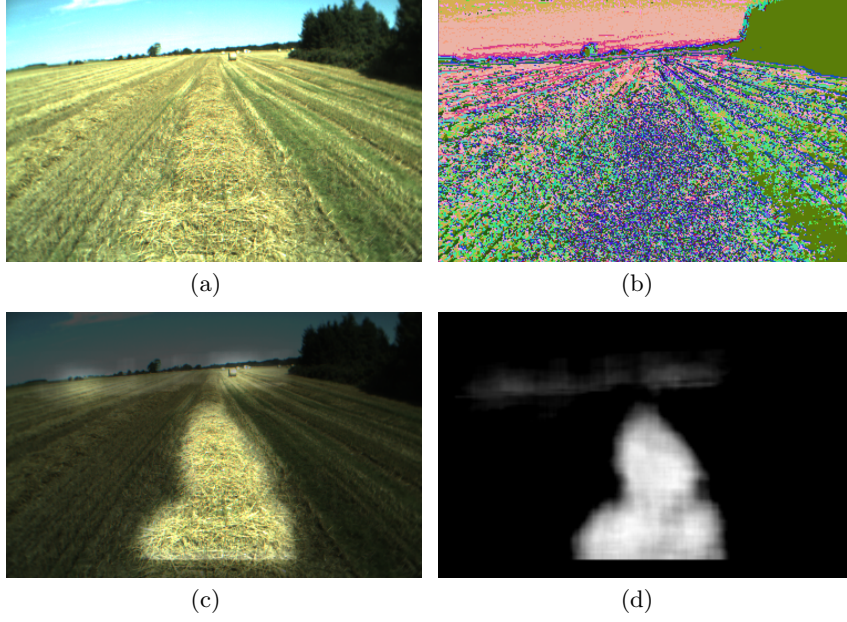


Fig. 9. (a) Left image from the stereo camera. (b) Texton classification. Pixel colour shows class of texton. (c) The stereo camera image with a transparency mask based on the swath classification. (d) Swath classification based on texture with intensity representing the strength of classification.

where $[l_c, a_c, b_c]$ is the colour of the pixel at this location in CIE*LAB colour-space. $(l_1 - l_c), \dots, (l_8 - l_c)$ are the intensity differences between the pixel at this location and the 8 surrounding pixels in a 3×3 neighbourhood. The vector elements are then weighted using $\{w_1 = 0.5, w_2 = 1, w_3 = 0.5\}$. A K-means algorithm is then run on all these descriptors to extract cluster centres which we refer to as textons. The fast K-means algorithm ([57]) was used in [13] to find the set of textons v_j that partitions the descriptors into κ sets $\mathcal{T} = \mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_\kappa$ by trying to minimise:

$$\mathcal{T}_{reg} = \underset{\mathcal{T}}{\operatorname{argmin}} \sum_{j=1}^{\kappa} \sum_{p_i \in \mathcal{T}_j} \|p_i - v_j\|^2 \quad (30)$$

As a final step, processing of all labelled pixels in an image at time k gives an estimate of swath density using the textons that are classified as belonging to the swath \mathcal{Y}_{swath} ,

$$s_t(k) = \max_{density} \mathcal{T}_{reg}(p_i \in \mathcal{Y}_{swath}, i = k - N \dots k) \quad (31)$$

Each pixel location in the image is then labelled by finding the nearest texton in Euclidean space. An example of the result of such a classification is shown in Fig.9.

One way of abstracting the texture processing could be in the form of a spline function describing the texture-classified swath,

$$\mathbf{s}_t(k) = \min_{spline} (s_t(j), j = k - N, k) \quad (32)$$

Another use of the texton classification could be to robustify the stereo estimate Eq. 27 by,

$$[s_{ct}, g_{ct}, h_{ct}]^T = \mathcal{V}_{reg}(i_l, i_r, \mathcal{T}_{reg}). \quad (33)$$

Using the *intelligent sensor* capability of Eq. 33 represents a fault-tolerant processing within the stereo algorithms that could be used after learning of the set \mathcal{T}_{swath} has been obtained. This is an alternative to using Eqs. 32 and 28 separately, but the latter could have benefits during unsupervised learning.

5.5 Structural Model

The constraints in this case describe the structured natural environment of the field with swath, the baler and the available sensors. Let the constraints be composed of those from measurements (m), differential (d), and the "system" constraints (c). Using \mathbf{v}^b for visual odometry measured velocity vector over ground seen in body coordinates; \mathbf{a}^b for IMU measured acceleration vector; \mathbf{p}^n the position in (North, East) coordinates with \mathbf{p}_1^n being the position measurement from the GPS; \mathbf{R}_b^n is the rotation matrix from body to navigation frame, which is a function of Θ , the attitude vector (Euler angles roll, pitch and yaw) and of λ , the latitude; s^b is the swath position spline seen in body coordinates. In this analysis, the \mathbf{R}_b^n matrix is assumed to be known.

With variables defined above, the sets of known and unknown variables are,

$$\begin{aligned} \mathcal{K} &= \{\mathbf{v}^b, \mathbf{a}^b, \mathbf{p}_1^n, s_g, s_c, s_m, s_t, \mathbf{R}_b^n(\Theta, \lambda)\} \\ \mathcal{X} &= \{\mathbf{p}^n, \dot{\mathbf{p}}^n, s^b, s, \varepsilon_x\} \end{aligned} \quad (34)$$

The constraints are summarised in Table 3.

Table 3. Behaviours for Baler Control Example

$$\begin{aligned} c_1 : s^b &= \mathbf{R}_b^n(\Theta, \lambda)s + \mathbf{p}^n \\ c_2 : \varepsilon_x &= \mathcal{E}(s^b) \\ d_1 : \dot{\mathbf{p}}^n &= \frac{d}{dt}\mathbf{p}^n \\ m_1 : \mathbf{v}^b &= \mathbf{R}_b^n(\Theta, \lambda)\dot{\mathbf{p}}^n \\ m_2 : \mathbf{a}^b &= \frac{d}{dt}\mathbf{R}_b^n(\Theta, \lambda)\dot{\mathbf{p}}^n \\ m_3 : \mathbf{p}_1^n &= \mathbf{p}^n \\ m_4 : s_g &= s \\ m_5 : s_c &= s^b \\ m_6 : s_m &= s^b \\ m_7 : s_t &= s^b \end{aligned}$$

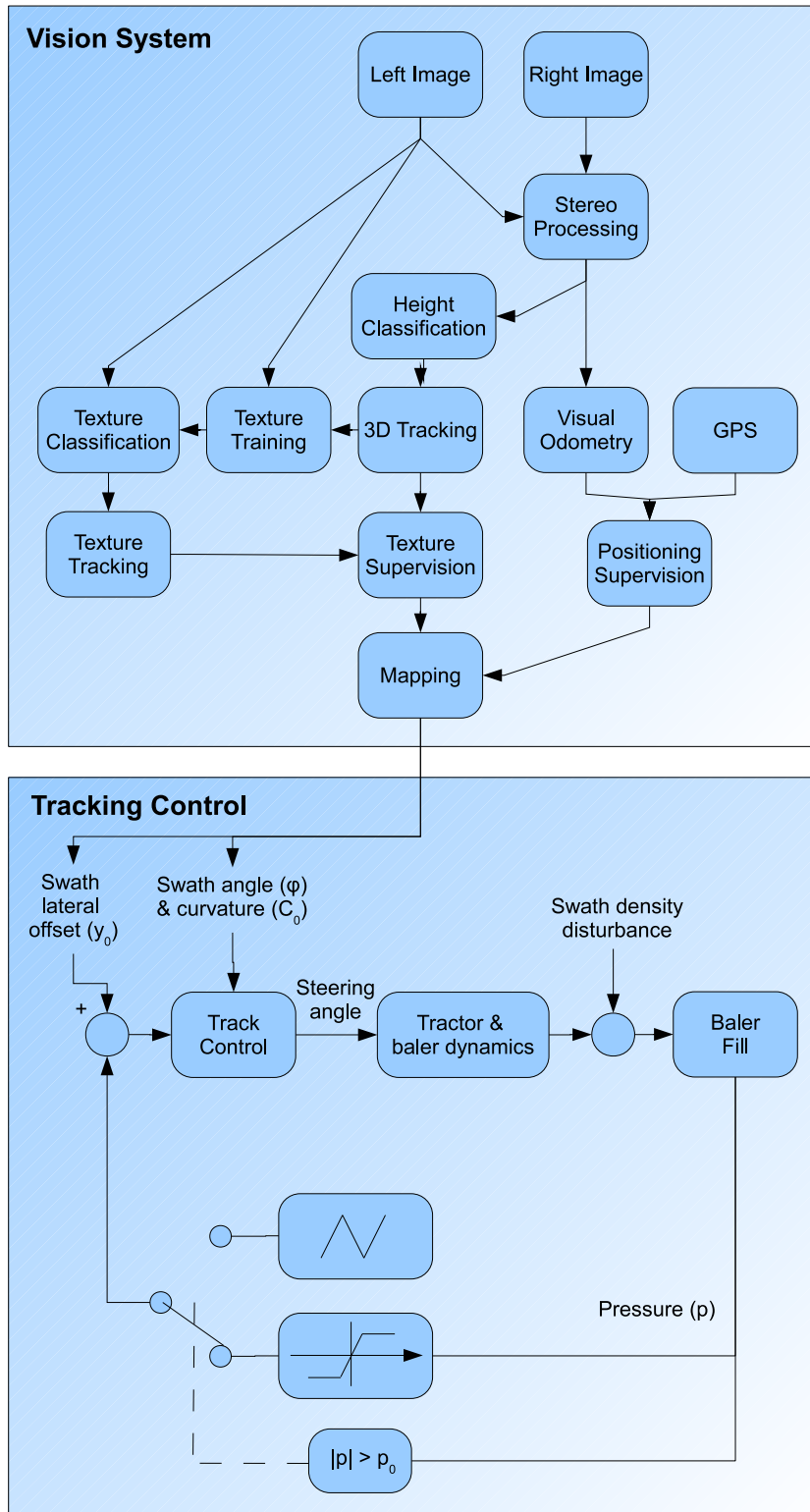


Fig. 10. Block diagram of vision system and tracking control for baling. Supervised classification and positioning provides mapping of field structures, which are fed to the steering controller.

5.6 Residuals for fault diagnosis

The number of constraints available depends on: the *use mode*, eg. m_7 is only available when the tractor is in the field with a swath and texture learning has been completed; the *configuration*, eg. whether an inertial measurement unit (IMU) is mounted; on *faults*, eg. failure of an instrument. Structural analysis under different use modes was presented in [67] and aspects of analysis under simultaneous faults was discussed in [11]. A structural analysis of the baler example, without the texture element, was presented in [12]. Analysing the constraints listed in Eq. 3, the following residuals are obtained,

$$\begin{aligned}
r_1 &= \mathbf{v}^b - \mathbf{R}_b^n(\Theta, \lambda) \frac{d}{dt} \mathbf{p}_1^n \\
r_2 &= \frac{d}{dt} \mathbf{v}^b - \mathbf{a}^b \\
r_3 &= \mathcal{E}(s_c) - \mathcal{E}(\mathbf{R}_b^n(\Theta, \lambda) s_g + \mathbf{p}_1^n) \\
r_4 &= \mathcal{E}(s_c) - \mathcal{E}(s_m) \\
r_5 &= \mathcal{E}(s_m) - \mathcal{E}(s_t)
\end{aligned} \tag{35}$$

5.7 Field Tests

The properties of residuals were investigated based on recorded data. The position of the swath was first logged by following the middle of the swath manually - emulating the vehicle forming the swath. This was repeated for a second pass emulating the vehicle that should pick up the swath. This provides some form of limited ground truth. The position error of the driver is bounded between the runs as he constantly steers relative to the swath.

Experience with driving with balers puts the error associated with not driving exactly over the center of the swath to under $\pm 0.2\text{m}$ as this is required to pick up the swath successfully. In the data examined the GPS has a false offset in the second pass relative to the first pass of approximately 0.6m for the first approx. 70s before it corrects its position estimate to bring it to about 0.15m of the swath location. This offset is acceptable for normal operation. Field tests enabled calculation of residuals r_1, r_3 and r_4 as an instrumentation issue prevented data reception from the IMU. The field test is hence representing a case of one permanent failure and an additional fault occurring.

The driver interfaces to the control system through a terminal to change settings and engages/disengages the automatic steering system through a switch. The baler has integrated pressure sensors which are used to measure the bale diameter. This information is used in the controller to assure an even filling of the bale chamber. A wheel angle sensor provides feedback about the angle of the front wheels relative to the tractor. Hydraulics allow actuation of the front wheels.

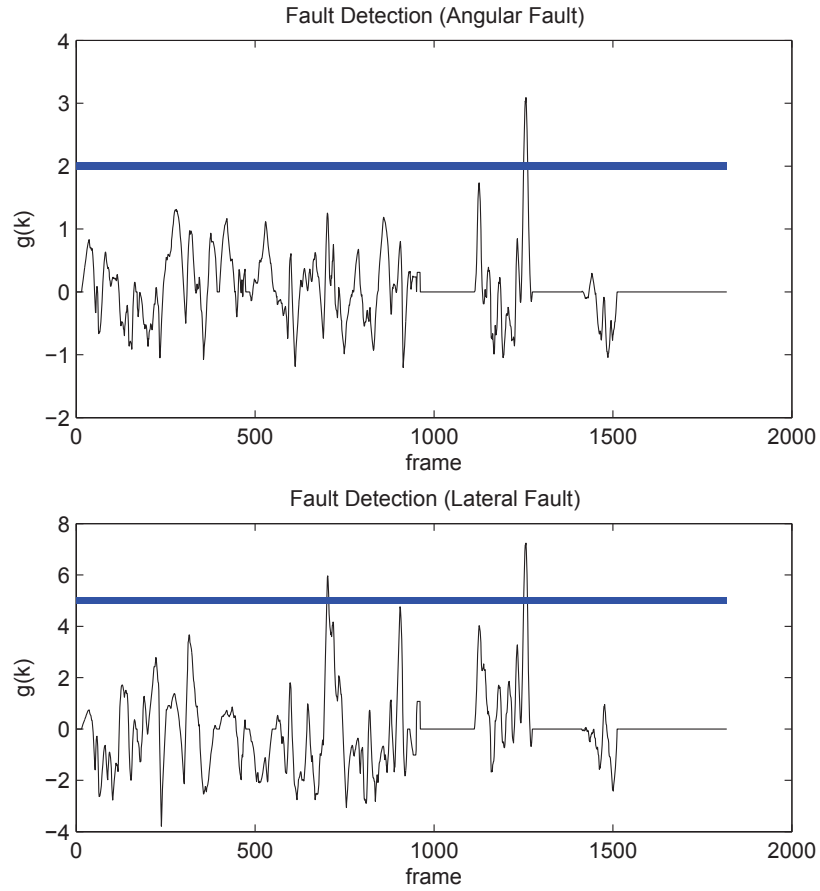


Fig. 11. Detection of deviation in angle or across the swath track using a CUSUM test. The thick line defines the threshold.

5.8 Control

Tests were conducted with a tracking control system that was made to collect the swath if the match score was above a predefined threshold. The control system remains active as long as map information is available ahead of the vehicle. A main difficulty in baling is that the bale chamber must be filled evenly. Pressure sensors inside the baler provide a measure of how evenly it is filled. If the bale chamber is unevenly filled then the bale becomes cone shaped. The bale must have a certain size before the pressure sensors give usable feedback. To compensate for this lack of feedback the controller has two states. In the initial state where pressure has not yet built up an open-loop steering pattern is followed where the vehicle changes between being the left and right edges of the swath. The steering system changes mode when the bale size reaches a threshold and sensor feedback is used. The sensor feedback spans an interval from -1

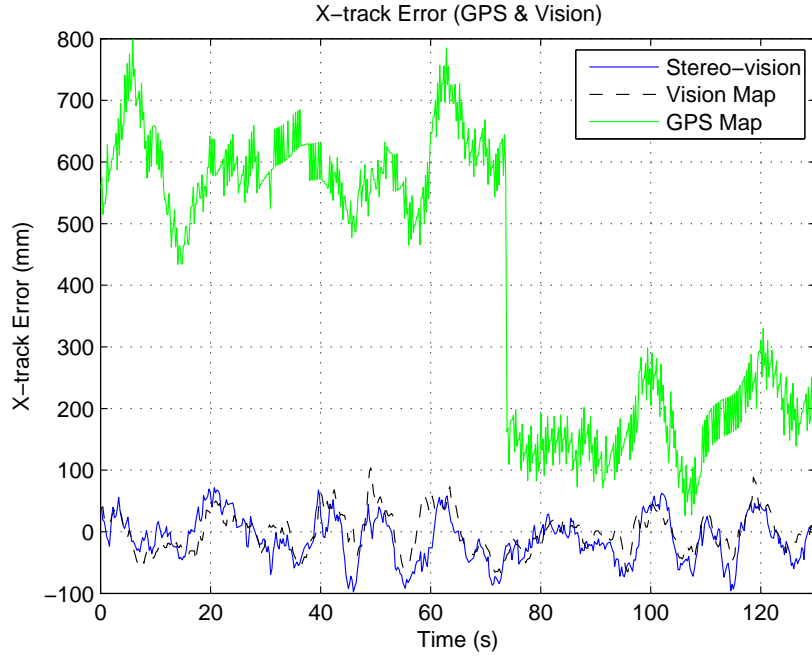


Fig. 12. The vehicle was driven manually over a swath. The driver centered the vehicle over the middle of the swath and drove for 2 min while maintaining this centered position. The x-track errors from the subsystems were recorded.

to 1 indicating how cone shaped the bale is. This information and the relative location of the swath (computed from a fused spline map) is used for control. The swath parameters y_0 , ϕ , and C_0 of the spline relative to the vehicle are then fed to a generic curve tracking controller. Fault-tolerant control described in this example focused on the complex parts, ie. the natural environment. Extension with the baling control part would be simple.

Figure 11 shows detection of deviations in steering angle and cross-track based on a CUSUM detection. The statistical change detection is much more robust and has less nuisance from noise than any of the single measurements.

6 Conclusions

This chapter has introduced to generic principles of diagnosis and fault-tolerant control based on the formal description of *behaviours* and *services*. The steps of system analysis, residual generation and change detection were discussed and exemplified through two large case studies. One dealt with a UAV sensor fault diagnosis and fault-handling, another with vision-based baling for agriculture.

It was shown that the generic theoretic methods are indeed applicable to the complexity met in a natural environment, and the chapter showed how such generalisation was obtained in both theory and in practice.

Acknowledgements

Research support from CLAAS Agrosystems, from the Danish Forces Joint UAV Team and the Danish Ministry for Science, Development and Innovation are gratefully acknowledged.

References

1. Åström, K.J., Albertos, P., Blanke, M., Isidori, A., Schaufelberger, W., Sanz, R. (eds.): Control of Complex Systems. Springer (2000)
2. Basseville, M., Nikiforov, I.V.: Detection of Abrupt Changes: Theory and Application. Information and System Science, Prentice Hall, New York (1993)
3. Basseville, M.: Detecting changes in signals and systems - a survey. *Automatica* 24(3), 309–326 (1988)
4. Bateman, F., Noura, H., Ouladsine, M.: Fault tolerant control strategy based on the doa: Application to uav. In: 7th IFAC Symposium on Fault Detection Supervision and Safety of Technical Processes (2009)
5. Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M.: Diagnosis and Fault-tolerant Control. Springer-Verlag, 2nd edn. (2003, 2nd ed 2006)
6. Blanke, M., Lorentzen, T.: Satool - a software tool for structural analysis of complex automation systems. In: 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes SAFEPROCESS. Beijing (2006)
7. Blanke, M.: Consistent design of dependable control systems. *Control Engineering Practice* 4(9), 1305–1312 (1996)
8. Blanke, M.: Fault-tolerant sensor fusion with an application to ship navigation. In: Proc. IEEE Mediterranean Control Conference. pp. 1385–1390 (June 2005)
9. Blanke, M., Izadi-Zamanabadi, R., Bøgh, S.A., Lunau, C.P.: Fault-tolerant control systems - a holistic view. *Control Engineering Practice* 5(5), 693–702 (1997)
10. Blanke, M., Izadi-Zamanabadi, R., Lootsma, T.: Fault monitoring and re-configurable control for a ship propulsion plant. *Journal of Adaptive Control and Signal Processing* 12, 671–688 (1998)
11. Blanke, M., Staroswiecki, M.: Structural design of systems with safe behavior under single and multiple faults. In: 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes SAFEPROCESS. pp. 511–516. Beijing, P. R. China (September 2006)
12. Blas, M.R., Blanke, M.: Natural environment modeling and fault-diagnosis for automated agricultural vehicle. In: Proceedings 17th IFAC World Congress, Seoul, Korea. pp. 1590–1595 (2008)
13. Blas, M.R., Blanke, M.: Automatic baling using stereo vision and texture learning. *Computers and Electronics in Agriculture* 75, 159–168 (2011)
14. Blas, M.R., Blanke, M., Madsen, T.E.: A method of detecting a structure in a field, a method of steering an agricultural vehicle and an agricultural vehicle. Patent (March 2010), eP10157313.7

15. Blas, M.R., Blanke, M., Rusu, R.B., Beetz, M.: Fault-tolerant 3d mapping with application to an orchard robot. In: Proc. 7th IFAC International Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS'09). pp. 893–898 (2009)
16. Blesa, J., Puig, V., Saludes, J.: Robust fault detection using polytope-based set-membership consistency test. IET Control Theory and Applications 6 (12), 1767–1777 (2012)
17. Caponetti, F., Blas, M.R., Blanke, M.: Stochastic automata for outdoor semantic mapping using optimised signal quantisation. Control Engineering Practice 19, 223–233 (2011)
18. Chen, J., Patton, R.J.: Optimal filtering and robust fault-diagnosis of stochastic-systems with unknown disturbance. IEE Proceedings Control Theory and Applications 143(1), 31–36 (1996)
19. Chen, J., Patton, R.J.: Robust Model-based Fault Diagnosis for Dynamic Systems. Kluwer Academic Publishers (1999)
20. Chow, E.Y., Willsky, A.S.: Analytical redundancy and the design of robust failure detection systems. IEEE Transactions on Automatic Control AC-29(7), 603–614 (1984)
21. Ding, S.M.: Model-based fault diagnosis techniques: design schemes, algorithms, and tools. Springer (2008)
22. Ducard, G.J.J.: Fault-tolerant Flight Control and Guidance Systems. Springer Verlag (2009)
23. Düşteğör, D., Cocquempot, V., Staroswiecki, M.: Structural analysis for fault detection and identification: an algorithmic study. In: Proceedings of the 2nd Symposium on System Structure and Control 2004 (SSSC'04). Oaxaca, Mexico (8-10 december 2004)
24. Düşteğör, D., Frisk, E., Cocquempot, V., Krysander, M., Staroswiecki, M.: Structural analysis of fault isolability in the damadics benchmark. Control Engineering Practice 14 (6), 597–608 (2006)
25. Dulmage, A.L., Mendelsohn, N.S.: A structure theory of bi-partite graphs. Trans. Royal Society of Canada. Sec. 3. 53, 1–13 (1959)
26. Dulmage, A.L., Mendelsohn, N.S.: Two algorithms for bipartite graphs. Journal of the Society for Industrial and Applied Mathematics. 11(1), 183–194 (1963)
27. Dunia, R., Qin, S.J., Edgar, T.F., McAvoy, T.J.: Identification of faulty sensors using principal component analysis. AIChE J 42, 2797–2812 (1996)
28. Düşteğör, D.: Structural Analysis for Fault Detection and Identification: Algorithmic Issues. Ph.D. thesis, Universite des Sciences et Technologies de Lille - France, Automatic, Computer and System Eng. (December 2005)
29. Emami-Naeini, A., Akhter, M., Rock, S.M.: Effect on model uncertainty on failure detection ; the threshold selector. IEEE Transactions on Automatic Control, Vol AC-33 no.12 pp. 1106–1115 (1988)
30. Feki, M.E., Jardin, A., Marquis-Favre, W., Krähenbühl, L., Thomasset, D.: Structural analysis by bond graph approach: duality between causal and bicausal procedures. Journal of Systems and Control Engineering 226, 82 – 100 (2012)
31. Fouladirad, M., Freitag, L., Nikiforov, I.: Optimal fault detection with nuisance parameters and a general covariance matrix. International Journal of Adaptive Control and Signal Processing 22 (5), 431 – 439 (2008)
32. Frank, P.M.: Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy. Automatica 26(3), 459–474 (1990)
33. Frank, P.M.: On-line fault detection in uncertain nonlinear systems using diagnosticobservers; a survey. Int. Journal of Systems Science 25(12), 2129–2154 (1994)

34. Frank, P.M., Ding, S.X., Marcu, T.: Model-based fault diagnosis in technical processes. *Transac. of the Institute of Measurement and Control* 22 (1), 57–101 (2000)
35. Frank, P.M., Ding, X.: Frequency domain approach to optimally robust residual generation and evaluation for model-based fault diagnosis. *Automatica* 30 (5), 789–804 (1994)
36. Fravolini, M.L., Brunori, V., Campa, G., Napolitano, M.R., La Cava, M.: Structured analysis approach for the generation of structured residuals for aircraft fdi. *IEEE Transac. on Aerospace and Electronic Systems* 45 (4), 1466–1482 (2009)
37. Freddi, A., Longhi, S., Monteriu, A.: A model-based fault diagnosis system for unmanned aerial vehicles. In: 7th IFAC Symp. on Fault Detection, Supervision and Safety of Technical Processes (2009)
38. Frisk, E., Krysander, M., Nyberg, M., Åslund, J.: A toolbox for design of diagnosis systems. In: *Proc. IFAC Safeprocess'2006* (2006)
39. Galeazzi, R., Blanke, M., Poulsen, N.K.: Early detection of parametric roll resonance on container ships. *IEEE Transac. of Control Systems Technology* 21 (2), 489 – 503 (2013)
40. Gertler, J.J.: Survey of model-based failure detection and isolation in complex plants. *IEEE Control Systems Magazine* 8(6), 3–11 (1988)
41. Gertler, J.J.: Fault detection and isolation using parity relations. *Control Engineering Practice* 5(5), 653–661 (1997)
42. Gertler, J.: *Fault detection and diagnosis in engineering systems*. Marcel Dekker (1998)
43. Gheorghe, A., Zolghadri, A., Cieslak, J., Goupil, P., Dayre, R., Berre, H.L.: Model-based approaches for fast and robust fault detection in an aircraft control surface servo loop: From theory to flight tests. *IEEE Control Systems Magazine* 33 (3), 20–30 (2013)
44. Gálvez-Carrillo, M., Kinnaert, M.: Sensor fault detection and isolation in doubly-fed induction generators accounting for parameter variations. *Renewable Energy* 36, 1447–1457 (2011)
45. Goshtasby, A.A.: *2-D and 3-D Image Registration: for Medical, Remote Sensing, and Industrial Applications*. Wiley (2005)
46. Goupil, P.: Airbus state of the art and practices on fdi and ftc in flight control systems. *Control Engineering Practice* 19(6), 524–539 (2011)
47. Gustafsson, F.: *Adaptive Filtering and Change Detection*. Wiley (2000 (2nd ed 2001))
48. Hansen, S., Blanke, M.: Diagnosis of airspeed measurement faults for unmanned aerial vehicles. *IEEE Transactions on Aerospace and Electronic Systems* 50 (1) (2014)
49. Hansen, S., Blanke, M., Adrian, J.: Diagnosis of uav pitot tube defects using statistical change detection. In: 7th IFAC Symposium on Intelligent Autonomous Vehicles. vol. 7(1). IFAC PapersOnLine (2010)
50. Heredia, G., Ollero, A., Bejar, M., Mahtani, R.: Sensor and actuator fault detection in small autonomous helicopters. *Mechatronics* 18, 90–99 (2008)
51. Hopcroft, J.E., Karp, R.M.: An $n^{\frac{5}{2}}$ algorithm for maximal matchings in bipartite graphs. *SIAM Journal on Computing* pp. 225–231 (1973)
52. Horak, D.T.: Failure detection in dynamic systems with modeling errors. *Journal of Guidance, Control and Dynamics* 11(6), 508–516 (1988)
53. Ingimundarson, A., Bravo, J.M., Puig, V., Alamo, T., Guerra, P.: Robust fault detection using zonotope-based set-membership consistency test. *Int. J. Adapt. Control Signal Processing* 23 (4), 311–330 (2009)

54. Isermann, R.: On fuzzy logic applications for automatic control, supervision, and faultdiagnosis. *IEEE Trans. On Sys. Man and Cyber. Part A-Sys. & Humans* 28(2), 221–235 (1998)
55. Isermann, R.: Supervision, fault-detection and fault-diagnosis methods - an introduction. *Control Engineering Practice* 5(5), 639–652 (May 1997)
56. Isermann, R.: *Fault-diagnosis systems: An introduction from fault detection to fault tolerance*. Springer (2005)
57. Jain, A., Dubes, R.: *Algorithms for Clustering Data*. Prentice-Hall, Englewood Cliffs (1988)
58. Kay, S.M.: *Fundamentals of Statistical Signal Processing: Detection Theory*. Prentice-Hall PTR (1998)
59. Kinnaert, M.: Robust fault detection based on observers for bilinear systems. *Automatica* 35, 1829–1842 (1999)
60. Knüppel, T., Blanke, M., Østergaard, J.: Fault diagnosis for electrical distribution systems using structural analysis. *Int. Journal of Robust and Nonlinear Control* (2014)
61. Korbicz, J.: Fault diagnosis of non-linear dynamical systems using analytical and soft computing methods. *Journal of Automation, Mobile Robotics & Intelligent Systems* 1, 7–23 (2007)
62. Korbicz, J., Koscielny, J.M., Kowalczyk, Z.: *Fault diagnosis: models, artificial intelligence, applications*. Springer (2004)
63. Krysander, M.: *Design and Analysis of Diagnosis Systems Using Structural Methods*. Ph.D. thesis, Linköping University (June 2006)
64. Krysander, M., Åslund, J., Nyberg, M.: An efficient algorithm for finding minimal overconstrained subsystems for model-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans* 38(1), 197 – 206 (2008)
65. Krysander, M., Frisk, E.: Leakage detection in a fuel evaporative system. *Control Engineering Practice* 17(11), 1273–1279 (2009)
66. Krysander, M., Nyberg, M.: Predicting fault isolability properties using structural and analytical information. In: *Proc. IFAC World Congress* (2005)
67. Laursen, M., Blanke, M., Düstegör, D.: Fault diagnosis in a water for injection system using enhanced structural isolation. *International Journal of Applied Mathematics and Computer Science* 18 (4), 593–603 (2008)
68. Leitold, A., Hangos, K.M.: Structural solvability analysis of dynamic process models. *Computers and Chemical Engineering* 25, 1633–1646 (2001)
69. Lunze, J.: Diagnosis of quantised systems. *Fault Detection, Supervision and Safety for Technical Processes 2000* 1(1), Elsevier Sci (June 2001)
70. Lunze, J., Schröder, J.: Process diagnosis based on a discrete–event description. *Automatisierungstechnik* 47(8) (1999)
71. Mahmoud, M.M., Jiang, J., Zhang, Y.: *Active fault tolerant control systems: stochastic analysis and synthesis*. Springer (2003)
72. Mangoubi, R.S.: *Robust Estimation and Failure Detection*. Springer-Verlag (1998)
73. Niemann, H.H., Stoustrup, J.: An architecture for fault-tolerant controllers. *International Journal of Control* 78 (14), 1091–1110 (2005)
74. Niemann, H.: A setup for active fault diagnosis. *IEEE Trans. on Automatic Control* 51(9), 1572–1578 (September 2006)
75. Nikiforov, I.V.: Two strategies in the problem of change detection and isolation. *IEEE Transactions on Information Theory* 43 (2), 770–776 (1997)
76. Noura, H., Theilliol, D., Ponsart, J.C., Cham, A.: *Fault-tolerant Control Systems - design and Practical Applications*. Springer (2009)

77. Park, W., Lee, S.H., Song, J.: Fault detection and isolation of durumi-ii using similarity measure. *Journal of Mechanical Science and Technology* 23, 302–310 (2009)
78. Patan, K., Witczak, M., Korbicz, J.: Towards robustness in neural network based fault diagnosis. *Int. J. Appl. Math. Comput. Sci* 18 (4), 443–454 (2008)
79. Patton, R.J.: Fault detection and diagnosis in aerospace systems using analytical redundancy. *IEE Computing & Control Eng. J.* 2(3), 127–136 (1991)
80. Patton, R.J., Frank, P.M., Clark, R.N. (eds.): *Fault Diagnosis in Dynamic Systems, Theory and Application*. Prentice Hall (1989)
81. de Persis, C., Isidori, A.: A geometric approach to non-linear fault detection and isolation. *IEEE Transactions on Automatic Control* 45 (6), 853–865 (2001)
82. Pivano, L., Johansen, T.A., Smogeli, O.N.: A four-quadrant thrust estimation scheme for marine propellers: Theory and experiments. *IEEE Transactions on Control Systems Technology* 17 (1), 215–226 (2009)
83. Poulsen, N.K., Niemann, H.: Active fault diagnosis based on stochastic tests. *International Journal of Applied Mathematics and Computer Science* 18 (4), 487–496 (2008)
84. Quevedo, J., Puig, V., Cembrano, G., Blanch, J., Aguilar, J., Saporta, D., Benito, G., Hedo, M., Molina, A.: Validation and reconstruction of flow meter data in the barcelona water distribution network. *Control engineering Practice* 18 (6), 640–651 (2010)
85. Romano, D., Kinnaert, M.: Robust design of fault detection and isolation systems. *Quality and Reliability Engineering International* 22 (5), 527–538 (2006)
86. Seydou, R., Raissi, T., Zolghadri, A., Efimov, D.: Actuator fault diagnosis for flat systems: A constraint satisfaction approach. *International Journal of Applied Mathematics and Computer Science* 23 (1), 171–181 (2013)
87. Staroswiecki, M., Attouche, S., Assas, M.L.: A graphic approach for reconfigurability analysis. In: *Proc. DX'99* (June 1999)
88. Staroswiecki, M., Cassar, J.P., Cocquempot, V.: Generation of optimal structured residuals in the parity space. In: *12th IFAC World Congress*. vol. 8, pp. 299–305 (Jul 1993)
89. Staroswiecki, M., Declerck, P.: Analytical redundancy in nonlinear interconnected systems by means of structural analysis. In: *Proc. IFAC AIPAC'89 Symposium*. vol. 2, pp. 23–27. Elsevier - IFAC (1989)
90. Staroswiecki, M., Gehin, A.L.: Control, fault tolerant control and supervision problems. In: *IFAC Safeprocess' 2000*. Budapest, Hungary (2000)
91. Stevens, B.L., Lewis, F.L.: *Aircraft Control and Simulation*. John Wiley & Sons, 2nd edn. (2003)
92. Stoorvogel, A.A., Niemann, H.H., Saberi, A., Sannuti, P.: Optimal fault signal estimation. *International Journal of Robust and Nonlinear Control* 12 (8), 697–727 (2002)
93. Svärd, C., Nyberg, M.: Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans* 40(6), 1310–1328 (2010)
94. Tabatabaeipour, S.M.: Active fault detection and isolation of discrete-time linear time-varying systems: A set-membership approach. *International Journal of Systems Science* (2014)
95. Travé-Massuyès, L., Escobet, T., Olive, X.: Diagnosability analysis based on component supported analytical redundancy relations. *IEEE Trans. on Systems, Man and Cybernetics, Part A : Systems and Humans* 36(6), 1146–1160 (2006)

96. Unger, J., Kröner, A., Marquardt, W.: Structural analysis of differential-algebraic equation systems - theory and applications. *Computers and Chemical Engineering* 19(8), 867–882 (1995)
97. Varga, A.: A numerically reliable approach for the synthesis of periodic fdi filters. In: *Proc. IFAC SAFEPROCESS 2012* (2012)
98. Varma, M., Zisserman, A.: Texture classification: Are filter banks necessary? *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 2, 691–696 (2003)
99. Willersrud, A., Blanke, M., Imsland, L., Pavlov, A.: Fault diagnosis of down-hole drilling incidents using adaptive observers and statistical change detection. *Journal of Process Control* (2015)
100. Wu, N.E., Klir, G.J.: Optimal redundancy management in reconfigurable control systems based on normalised nonspecificity. *Int. Journal of Systems Science* 31, 797–808 (2000)
101. Zhang, X.: *Auxiliary signal design in fault detection and diagnosis*. Springer Verlag (1989)
102. Zhou, W.-w., Blanke, M.: Identification of a class of non-linear state space models using rpe techniques. *IEEE Transac. of Automatic Control* 34 (3), 312–316 (1989)
103. Zolghadri, A.: Advanced model-based fdir techniques for aerospace systems: Today challenges and opportunities. *Progress in Aerospace Sciences* 53, 18–29 (2012)